

Capítulo 1: Introdução à optimização

Ana I. Pereira[‡]

[‡] Instituto Politécnico de Bragança, Portugal, apereira@ipb.pt

Matemática Aplicada

Problema

Seja f é uma função não linear, tal que, $f : \mathbb{R}^n \rightarrow \mathbb{R}$.
Pretende-se determinar o ponto $x^* \in \mathbb{R}^n$, tal que

$$\begin{array}{ll}\min & f(x) \\ \text{s.a} & c_j(x) = 0, j \in I \\ & c_k(x) \leq 0, k \in D \\ & l_i \leq x_i \leq u_i, i = 1, \dots, n\end{array}$$

Optimização sem/com restrições

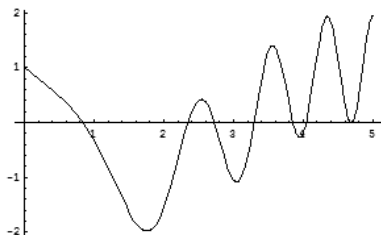


Figura : Optimização sem restrições

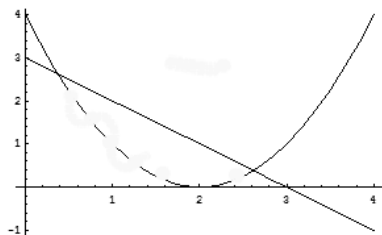


Figura : Optimização com restrições

Definições

Diz-se que :

- x representa o **vector das variáveis**;
- f representa a **função objectivo**;
- c representa o **vector das restrições**;
- I e D representam o **conjunto de índices** das restrições de igualdade e desigualdade, respectivamente.

Exemplo: otimização sem /com restrição

Exemplo 1:

$$\begin{array}{ll}\min & (x - 2)^2 \\ & x \in \mathbb{R}\end{array}$$

Exemplo 2:

$$\begin{array}{ll}\min & (x - 2)^2 \\ \text{s.a} & 3 \leq x \\ & -5 \leq x \leq 5\end{array}$$

Mais definições

Designa-se por:

- Optimização contínua - se o vector das variáveis pertence a $\mathbb{R}^n$;
- Optimização discreta - se o vector das variáveis pertence a um conjunto discreto (ex. $\mathbb{N}$, $\mathbb{Z}$);
- Optimização inteira mista - se algumas variáveis pertencem a $\mathbb{R}$ e as restantes pertencem a um conjunto discreto;
- Optimização local - se pretendemos uma qualquer solução do problema, isto é, $f(x^*) \leq f(x) \forall x \in \mathcal{V}_\delta(x^*)$ (com $\delta > 0$);
- Optimização global - se pretendemos determinar a solução com valor menor, isto é, $f(x^*) \leq f(x) \forall x \in \mathbb{R}^n$;

Métodos para a optimização

Os métodos de optimização existentes são iterativos. Em geral, necessitam de uma aproximação inicial e geram uma sucessão de aproximações à solução do problema. O método de optimização é caracterizado pelo algoritmo do método e pelo estudo da convergência do método.

Informações usadas pelos algoritmos de optimização:

- valores da função objectivo;
- valores do vector das restrições;
- valores da primeira derivada (função objectivo e/ou restrições);
- valores da segunda derivada (função objectivo e/ou restrições);
- valores gerados nas iterações anteriores (última iterações vs k -ésimas iterações).

Métodos para a otimização

Os métodos de otimização são caracterizados por:

- **Robustez** - o algoritmo deverá funcionar para uma grande variedade de problemas.
- **Eficiência** - o algoritmo não deve requerer muito tempo/memória para resolver o problema.
- **Precisão** - o algoritmo deverá determinar uma aproximação à solução com uma boa precisão.

Mínimo/Minimizante

Seja $x^* \in \mathbb{R}^n$ o ponto onde se verifica $f(x^*) \leq f(x)$ para $\forall x \in \mathcal{V}_\delta(x^*)$.

Diz-se:

- **Mínimo do problema** - valor $f(x^*)$;
- **Minimizante local do problema** - ponto x^* tal que $f(x^*) \leq f(x)$ para $\forall x \in \mathcal{V}_\delta(x^*)$ e $x_i \neq x_i^*$ para $i = 1, \dots, n$;
- **Minimizante global do problema** - ponto x^* tal que $f(x^*) \leq f(x)$ para $\forall x \in \mathbb{R}^n$.

Máximo/Mínimo

$$\max f(x) = -(\min -f(x))$$

$$-f(2) = \min (x - 2)^2$$

e

$$f(2) = \max -(x - 2)^2$$

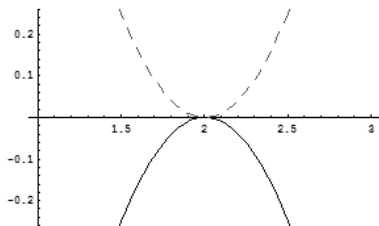


Figura : Maximização versus minimização

Referências



J. Nocedal e S. Wright. Numerical Optimization. *Springer Series in Operations Research*. Springer. 1999.



S. G. Nash e A. Sofer. Linear and Nonlinear Programming. *Industrial Engineering Series*. McGraw-Hill International Editions. 1996.



M. S. Bazaraa, H. D. Sherali e C. M. Shetty. Nonlinear Programming Theory and Algorithms. John Wiley and Sons. 1993.

Capítulo 2: Optimização sem restrições

Ana I. Pereira[‡]

[‡] Instituto Politécnico de Bragança, Portugal, apereira@ipb.pt

Matemática Aplicada

Formulação Matemática

Seja f é uma função não linear, tal que, $f : \mathbb{R}^n \rightarrow \mathbb{R}$.

Pretende-se determinar o ponto $x^* \in \mathbb{R}^n$, tal que

$$\min_{x \in \mathbb{R}^n} f(x)$$

Exemplos

Exemplo 1

$$\min_{(x_1, x_2) \in \mathbb{R}^2} (x_1 - 3)^2 + (x_2 - 2)^2$$

Exemplo 2

$$\min_{(x_1, x_2, x_3) \in \mathbb{R}^3} e^{(x_1+5)^2} + \ln(x_2^2 + 2) + x_3^2$$

Resolução analítica

Seja f continuamente diferenciável em $\mathbb{R}^n$.

- Passo 1: Resolver $\nabla f(x) = 0_{\mathbb{R}^n}$, obtendo os pontos estacionários $\bar{x}^{(i)} \in \mathbb{R}^n$ para $i = 1, \dots, K$
- Passo 2: Determinar $\nabla^2 f(\bar{x}^{(i)})$, para, $i = 1, \dots, K$

Resolução analítica

Seja $\bar{x}^{(i)}$, para $i = 1, \dots, K$, um ponto estacionário em $\mathbb{R}^n$, e f duas vezes continuamente diferenciável em $\mathcal{V}_\delta(\bar{x}_i)$, com $\delta > 0$.

Conclusões

- Se $\nabla^2 f(\bar{x}^{(i)})$ é definida positiva então $f(\bar{x}^{(i)})$ é mínimo local em $\mathcal{V}_\delta(\bar{x}^{(i)})$;
- Se $\nabla^2 f(\bar{x}^{(i)})$ é definida negativa então $f(\bar{x}^{(i)})$ é máximo local em $\mathcal{V}_\delta(\bar{x}^{(i)})$;
- Se $\nabla^2 f(\bar{x}^{(i)})$ é indefinida então $x^{(i)}$ é ponto sela em $\mathcal{V}_\delta(\bar{x}^{(i)})$;
- Se $\nabla^2 f(\bar{x}^{(i)})$ é semidefinida negativa (positiva) então nada se pode concluir.

Definições

Matriz definida positiva

Seja $A \in M_{n \times n}(\mathbf{R})$. A matriz A é definida positiva se:

- $x^T A x > 0$ para qualquer vector real $x \neq 0$.
- Os valores propios da matriz A são todos positivos.

Matriz definida negativa

Seja $A \in M_{n \times n}(\mathbf{R})$. A matriz A é definida negativa se:

- $x^T A x < 0$ para qualquer vector real $x \neq 0$.
- Os valores propios da matriz A são todos negativos.

Métodos iterativos

Os métodos iterativos para a resolução de problemas de optimização sem restrições podem-se dividir em:

- Métodos do gradiente;
- Métodos de procura directa.

Métodos do gradiente

Propriedades/Características

- usam informação da função e das derivadas (gradiente ou gradiente/Hessiana);
- só podem ser usados na resolução de problemas diferenciáveis;
- convergem rapidamente.

Algoritmos baseados na procura linear

Seja k uma dada iteração, e f é continuamente diferenciável.

Estes algoritmos determinam a nova aproximação $x^{(k+1)}$ a partir de $x^{(k)}$ e na direcção de procura $p^{(k)}$. A direcção de procura $p^{(k)}$ tem que ser uma direcção **descendente** para $x^{(k)}$ ($\nabla^T f(x^{(k)})p^{(k)} < 0$).

Assim, pretende-se resolver

$$\begin{array}{ll} \min & f(x^{(k)} + \alpha p_k) \\ \alpha > 0 \end{array} \quad (1)$$

Podemos resolver o problema (1) de forma exacta (muito dispendioso) ou de forma aproximada.

Algoritmo geral de método baseados na procura linear

Seja $k = 0$ e considere uma aproximação inicial $x^{(0)}$.

Algoritmo geral

- Passo 1: Calcular direcção de procura $p^{(k)}$
- Passo 2: Determinar $\alpha_k \in (0, 1]$ tal que

$$f(x^{(k)} + \alpha_k p^{(k)}) < f(x^{(k)})$$

- Passo 3: $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$
- Se não for satisfeito o critério de paragem ir para o Passo 1 com $k = k + 1$.

Método de Descida Máxima

Seja k uma dada iteração.

A direcção de procura é dada por

$$p^{(k)} = -\nabla f(x^{(k)})$$

- $-\nabla f(x^{(k)})$ é a direcção para onde $f(x)$ decresce mais rapidamente.
- Temos que determinar $\alpha > 0$ para determinar a nova aproximação à solução $x^{(k+1)}$.

Algoritmo do método de Descida Máxima

Seja $k = 0$ e considere uma aproximação inicial $x^{(0)}$.

Método de Descida Máxima

- Passo 1: $p^{(k)} = -\nabla f(x^{(k)})$
- Passo 2: Determinar $\alpha_k \in (0, 1]$ tal que

$$f(x^{(k)} + \alpha_k p^{(k)}) < f(x^{(k)})$$

- Passo 3: $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$
- Se não for satisfeito o critério de paragem ir para o Passo 1 com $k = k + 1$.

Método de Descida Máxima

Propriedades

Necessita das derivadas de 1ª ordem.

Principal desvantagem

O método é lento!!!

Ilustração

<http://www.cse.illinois.edu/iem/optimization/SteepestDescent/>

Método de Newton

Teorema de Taylor

Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continuamente diferenciável e seja $p \in \mathbb{R}^n$.
Então

$$f(x + p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x + tp) p$$

para algum $t \in (0, 1)$.

Seja x uma aproximação à solução. Usando o Teorema de Taylor, temos que

$$f(x + p) \approx f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x) p$$

ao vector p designamos por direcção de procura.

Método de Newton

Resolver

$$\min_{p \in \mathbb{R}^n} f(x + p)$$

Seja k uma dada iteração e $x^{(k)}$ uma aproximação à solução.
Usando o Teorema de Taylor, temos que

$$f(x^{(k)} + p) \approx f(x^{(k)}) + \nabla f(x^{(k)})^T p + \frac{1}{2} p^T \nabla^2 f(x^{(k)}) p$$

Algoritmo do Método de Newton

Seja $k = 0$ e considere uma aproximação inicial $x^{(0)}$.

Algoritmo do Método de Newton

- Passo 1: $p^{(k)} = -\nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)})$
- Passo 2: Determinar $\alpha_k \in (0, 1]$ tal que

$$f(x^{(k)} + \alpha_k p^{(k)}) < f(x^{(k)})$$

- Passo 3: $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$
- Se não for satisfeito o critério de paragem ir para o Passo 1 com $k = k + 1$.

Método de Newton

Propriedades

- Se $\nabla^2 f(x^{(k)})^{-1}$ for definida positiva então $p^{(k)}$ é uma direcção descendente (porquê?).
- Se $\nabla^2 f(x^{(k)})^{-1}$ for não definida positiva então $p^{(k)}$ pode não existir (porquê?).
- O Método de Newton tem convergência local.
- O Método de Newton tem convergência rápida quando $x^{(0)}$ está próxima da solução.

Principal desvantagem

Necessita de $\nabla^2 f(x^{(k)})^{-1}$ em cada iteração!!!

Método de Newton

Teorema

Se $x^{(0)}$ estiver suficientemente próximo de um mínimo local x^* e $\nabla^2 f(x^*)$ é definida positiva, então o método de Newton tem iterações bem definidas e converge quadraticamente para a solução x^* .

Ilustração

http://www.cse.illinois.edu/iem/optimization/Newton_Opt/

http://www.cse.illinois.edu/iem/optimization/Newton_Opt2D/

Método de Quasi-Newton

1ª Ideia

Substituir $\nabla^2 f(x^{(k)})$ por uma aproximação B_k calculada em cada iteração.

2ª Ideia

Substituir $\nabla^2 f(x^{(k)})^{-1}$ por uma aproximação H_k calculada em cada iteração.

Método de Quasi-Newton

Seja $k = 0$ e considere uma aproximação inicial $x^{(0)}$.

Algoritmo do Método de Quasi-Newton

- Passo 1: $p^{(k)} = -H_k \nabla f(x^{(k)})$
- Passo 2: Determinar $\alpha_k \in (0, 1]$ tal que

$$f(x^{(k)} + \alpha_k p^{(k)}) < f(x^{(k)})$$

- Passo 3: $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$
- Se não for satisfeito o critério de paragem ir para o Passo 1 com $k = k + 1$.

Método de Quasi-Newton

Técnicas para calcular H_k

- Actualização DFP (Davidon, Fletcher, Powell)
- Actualização BFGS (Broyden, Fletcher, Goldfarb, Shanno)
- Actualização SR1 quasi-Newton
- ...

Método de Quasi-Newton

Em cada iteração k , considere

$$s_k = x^{(k+1)} - x^{(k)} \text{ e } y_k = \nabla f(x^{(k+1)}) - \nabla f(x^{(k)})$$

Actualização DFP

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k}$$

Actualização BFGS

$$H_{k+1} = \left(I - \frac{s_k y_k^T}{y_k^T s_k} \right) H_k \left(I - \frac{y_k s_k^T}{y_k^T s_k} \right) + \frac{s_k s_k^T}{y_k^T s_k}$$

Método de Quasi-Newton

Nota

Se $y_k^T s_k > 0$ então as matrizes H_k são definidas positivas.

Método de Quasi-Newton

Condição 2.1

A matriz Hessiana, H , é Lipschitz continua em x^* , isto é,

$$\|H(x) - H(x^*)\| \leq L \|x - x^*\|,$$

para todo o x numa vizinhança x^* , com L um valor positivo.

Teorema

Seja f uma função duas vezes continuamente diferenciável e que a sucessão de aproximações converge para o minimizante x^* onde a condição 2.1 é satisfeita. Considere que também se verifica $\sum_{k=1}^{\infty} \|x^{(k)} - x^*\| \leq \infty$, para uma dada iteração k . Então $x^{(k)}$ converge para x^* superlinearmente.

Ilustração

Técnicas de globalização

Os métodos do gradiente (quando convergem) convergem para um ponto estacionário. Porquê implementar uma técnica de globalização?

- para garantir que o método converge para um ponto estacionário que é minimizante;
- para garantir que o método converge, qualquer que seja a aproximação inicial $x^{(0)}$ (isto é, $x^{(0)}$ pode estar fora da região de convergência).

Condição de Armijo

Dados $x^{(k)}$ e p_k , calcular $\alpha_k \in (0, 1]$ tal que satisfaz a seguinte condição

Condição de Armijo

$$f(x^{(k)} + \alpha_k p^{(k)}) \leq f(x^{(k)}) + \mu \alpha_k \nabla f(x^{(k)})^T p^{(k)}$$

Exercícios

Considere o seguinte problema

$$\min_{x \in \mathbb{R}^n} f(x) = 0.5x_1^2 \left(\frac{x_1^2}{6} + 1 \right) + x_2 \arctan(x_2) - 0.5 \ln(x_2^2 + 1)$$

- Resolva-o usando o método de descida máxima.
- Resolva-o usando o método de Newton.
- Resolva-o usando o método Quasi-Newton com actualização BFGS

Considere as aproximações iniciais $x^{(0)} = (1, 0.7)$ e $y^0 = (1, 2)$.

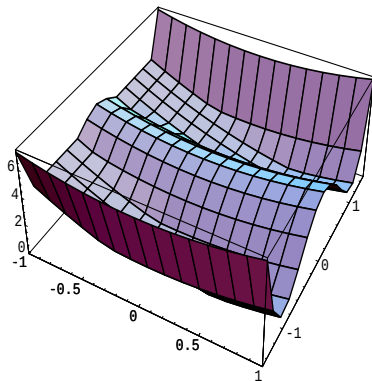
Método de procura directa

Propriedades

- Só usam informação da função objectivo;
- São apropriados para problemas não diferenciáveis (embora possam ser usados em problemas diferenciáveis);

Método de procura directa

$$\min f(x_1, x_2) \equiv \max\{(x_1 - 1)^2, x_1^2 + 4(x_2^2 - 1)\}$$



Método de Nelder-Mead

O método de Nelder-Mead define em cada iteração um **simplex**.

Em $\mathbb{R}^n$, sejam $x^1, x^2, \dots, x^n, x^{n+1}$ - $n + 1$ pontos, que representam os vértices do simplex de dimensão n .

Notação:

$$S_k = \langle x^1, x^2, \dots, x^n, x^{n+1} \rangle$$

representa o simplex, da iteração k , em que os vértices já estão ordenados por ordem crescente dos valores da função objectivo, isto é,

$$f(x^1) \leq f(x^2) \leq \dots \leq f(x^n) \leq f(x^{n+1}).$$

Método de Nelder-Mead

Em cada **iteração k** ,

definem-se pontos auxiliares - candidatos a vértices de um novo simplex - que serão aceites ou rejeitados de acordo com os correspondentes valores da função.

Método de Nelder-Mead

Operações básicas na construção dos pontos auxiliares:

- reflectir,
- expandir,
- contrair,
- encolher.

Método de Nelder-Mead

Estas operações são baseadas no **centróide**: $x^c = \frac{1}{n} \sum_{i=1}^n X^i$.

- Vértice reflectido: $x^r = 2x^c - X^{n+1}$.
- Vértice expandido: $x^e = 2x^r - x^c$.
- Vértice contraído para o exterior: $x^{ce} = \frac{1}{2}x^r + \frac{1}{2}x^c$.
- Vértice contraído para o interior: $x^{ci} = \frac{1}{2}X^{n+1} + \frac{1}{2}x^c$.
- Encolher o simplex - substituir cada um dos vértices X^i , $i = 2, \dots, n+1$, pelo ponto médio do segmento que une X^i a X^1 , ou seja, $x^i = \frac{X^i + X^1}{2}$ e formar simplex com $X^1, x^2, \dots, x^n, x^{n+1}$.

Método de Nelder-Mead

Seja $k = 1$ e considere um simplex S_1 .

Algoritmo do Método de Nelder-Mead

- Passo 1: Calcular o vértice reflectido x^r
- Passo 2: Se $f(X^1) \leq f(x^r) < f(X^n)$ aceitar x^r ($\bar{x} = x^r$) e ir para Passo 6
- Passo 3: Se $f(x^r) < f(X^1)$ calcular vértice expandido x^e
Se $f(x^e) < f(X^1)$ aceitar x^e ($\bar{x} = x^e$) e ir para Passo 6.
Senão aceitar x^r ($\bar{x} = x^r$) e ir para Passo 6.

Método de Nelder-Mead - continuação

Algoritmo do Método de Nelder-Mead - continuação

- Passo 4: Se $f(X^n) \leq f(x^r) < f(X^{n+1})$ então calcular vértice contraído para o exterior x^{ce}
Se $f(x^{ce}) < f(X^n)$ aceitar x^{ce} ($\bar{x} = x^{ce}$) e ir para Passo 6.
Senão encolher simplex.
- Passo 5: Se $f(x^r) \geq f(X^{n+1})$ então calcular vértice contraído para o interior x^{ci}
Se $f(x^{ci}) < f(X^n)$ aceitar x^{ci} ($\bar{x} = x^{ci}$) e ir para Passo 6.
Senão encolher simplex.
- Passo 6: Formar simplex S_{k+1} com $X^1, X^2, \dots, X^n, \bar{x}$.
- Passo 7: Se não for satisfeito o critério de paragem ir para o Passo 1 com $k = k + 1$.

Método de Nelder-Mead

Critério de paragem

Terminar o processo até o tamanho do simplex seja suficientemente pequeno.

$$\frac{1}{\Delta} \max_{2 \leq i \leq n+1} \|X^i - X^1\|_2 \leq \epsilon$$

onde $\Delta = \max\{1, \|X^1\|_2\}$ e $\epsilon > 0$.

Ilustração

<http://www.cse.illinois.edu/iem/optimization/NelderMead/>

Métodos para a optimização sem restrições

- Variantes do método dos gradientes conjugados.
- Trust region.
- Variantes do simulated annealing.
- Variantes do branch-and-bound.
- ...

Em Matlab

Toolbox - Optimization -

Esta Toolbox pertence ao software MatLab possuindo ferramentas para vários tipos de problemas na área de optimização:

- optimização sem restrições,
- optimização com restrições,
- optimização discreta.

Em Matlab

A toolbox inclui funções para resolver problemas da área:

- programação linear,
- programação quadrática,
- optimização não linear,
- mínimos quadrados,
- sistemas de equações não lineares,
- optimização multiobjectivo,
- programação inteira.

Em Matlab

Toolbox - Optimization (minimização)

- **bintprog** - Solve binary integer programming problems
- **fminbnd** - Find minimum of single-variable function on fixed interval
- **fminsearch** - Find minimum of unconstrained multivariable function using derivative-free method
- **fminunc** - Find minimum of unconstrained multivariable function
- **linprog** - Solve linear programming problems
- **quadprog** - Solve quadratic programming problems

Em Matlab

Toolbox - Optimization (minimização)

- **fmincon** - Find minimum of constrained nonlinear multivariable function
- **fgoalattain** - Solve multiobjective goal attainment problems
- **fminimax** - Solve minimax constraint problem
- **fseminf** - Find minimum of semi-infinitely constrained multivariable nonlinear function
- **ktrlink** - Find minimum of constrained or unconstrained nonlinear multivariable function using KNITRO third-party libraries

Em Matlab

Toolbox - Optimization (Sistemas de equações)

- **fsolve** - Solve system of nonlinear equations
- **fzero** - Find root of continuous function of one variable

Toolbox - Optimization (Mínimos quadrados)

- **lsqcurvefit** - Solve nonlinear curve-fitting (data-fitting) problems in least-squares sense
- **lsqlin** - Solve constrained linear least-squares problems
- **lsqnonlin** - Solve nonlinear least-squares (nonlinear data-fitting) problems
- **lsqnonneg** - Solve nonnegative least-squares constraint problem

Função: fminbnd

Seja $[a, b] \subset \mathbb{R}$.

$$\min_{a \leq x \leq b} fun(x)$$

Sintaxe:

`x = fminbnd(fun,a,b)`

`x = fminbnd(fun,a,b,options)`

`[x,fval] = fminbnd(...)`

`[x,fval,exitflag] = fminbnd(...)`

`[x,fval,exitflag,output] = fminbnd(...)`

Função: fminbnd

Variáveis de entrada:

- fun - M-file com a função objectivo
- a - limite inferior do intervalo
- b - limite superior do intervalo
- options - opções criadas com optimset

Função: fminbnd

Variáveis de saída:

- `exitflag` - inteiro que informa como o algoritmo pára.
 - 1 : A função converge para a solução.
 - 0 : O número de iterações excedeu `MaxIter` ou o número de avaliações da função excedeu `FunEvals`.
 - 1 : O algoritmo terminou com problemas na função objectivo.
 - 2 : Os limites do intervalo são inconsistentes.

Função: fminbnd

Variáveis de saída:

- output - estrutura com informações do processo de optimização.
 - iterations : Número de iterações.
 - funcCount : Número de avaliações da função objectivo.
 - algorithm : Algoritmo usado.
 - message : Mensagem de saída.

Função: fminsearch

Seja $x \in \mathbb{R}^n$.

$$\min fun(x)$$

Sintaxe:

`x = fminsearch(fun,x0)`

`x = fminsearch(fun,x0,options)`

`x = fminsearch(problem)`

`[x,fval] = fminsearch(...)`

`[x,fval,exitflag] = fminsearch(...)`

`[x,fval,exitflag,output] = fminsearch(...)`

Função: fminsearch

Variáveis de entrada:

- fun - M-file com a função objectivo
- x0 - ponto inicial
- options - opções criadas com optimset

Função: fminsearch

Variáveis de saída:

- `exitflag` - inteiro que informa como o algoritmo pára.
 - 1 : A função converge para a solução.
 - 0 : O número de iterações excedeu `MaxIter` ou o número de avaliações da função excedeu `FunEvals`.
 - 1 : O algoritmo terminou com problemas na função objectivo.

Função: fminsearch

Variáveis de saída:

- output - estrutura com informações do processo de optimização.
 - iterations : Número de iterações.
 - funcCount : Número de avaliações da função objectivo.
 - algorithm : Algoritmo usado.
 - message : Mensagem de saída.

Função: fminunc

Seja $x \in \mathbb{R}^n$.

$$\min fun(x)$$

Sintaxe:

`x = fminunc(fun,x0)`

`x = fminunc(fun,x0,options)`

`x = fminunc(problem)`

`[x,fval] = fminunc(...)`

`[x,fval,exitflag] = fminunc(...)`

`[x,fval,exitflag,output] = fminunc(...)`

`[x,fval,exitflag,output,grad] = fminunc(...)`

`[x,fval,exitflag,output,grad,hessian] = fminunc(...)`

Função: fminunc

Variáveis de entrada:

- fun - M-file com a função objectivo
- x0 - ponto inicial
- options - opções criadas com optimset

Função: fminunc

Variáveis de saída:

- `exitflag` - inteiro que informa como o algoritmo pára.
 - 1 : A magnitude do gradiente no ponto é menor que `TolFun`.
 - 2 : Alteração das últimas duas iterações é menor que `TolX`.
 - 3 : Alteração dos valores da função objectivo é menor que `TolFun`.
 - 5 : Decréscimo dos valores da função é menor que `TolFun`.
 - 0 : O número de iterações excedeu `MaxIter` ou o número de avaliações da função excedeu `FunEvals`.
 - 1 : O algoritmo terminou com problemas na função objectivo.

Função: fminunc

Variáveis de saída:

- `grad` - Magnitude do vector gradiente na aproximação.
- `hessian` : Valor da Hessiana na aproximação.

Função: fminunc

Variáveis de saída:

- output - estrutura com informações do processo de optimização.
 - iterations : Número de iterações.
 - funcCount : Número de avaliações da função objectivo.
 - firstorderopt : Medida da condição de 1ª ordem.
 - algorithm : Algoritmo usado.
 - cgiterations : Número total de iterações PCG (algoritmo de grande dimensão).
 - stepsize : Valor do último comprimento do passo.
 - message : Mensagem de saída.

Exercícios

Considere o seguinte problema

$$\min_{x \in \mathbb{R}^n} f(x) = 0.5x_1^2 \left(\frac{x_1^2}{6} + 1 \right) + x_2 \arctan(x_2) - 0.5 \ln(x_2^2 + 1)$$

- Resolva-o usando o método de Newton.
- Resolva-o usando o método Quasi-Newton com actualização BFGS e DFP.
- Resolva-o usando o método Nelder-Mead.

Considere as aproximações iniciais $x^{(0)} = (1, 0.7)$ e $y^{(0)} = (1, 2)$.

Referências



J. Nocedal e S. Wright. Numerical Optimization. *Springer Series in Operations Research*. Springer. 1999.



S. G. Nash e A. Sofer. Linear and Nonlinear Programming. *Industrial Engineering Series*. McGraw-Hill International Editions. 1996.



M. S. Bazaraa, H. D. Sherali e C. M. Shetty. Nonlinear Programming Theory and Algorithms. John Wiley and Sons. 1993.



www.mathworks.com

Capítulo 3: Optimização com restrições

Ana I. Pereira[‡]

[‡] Instituto Politécnico de Bragança, Portugal, apereira@ipb.pt

Matemática Aplicada

Formulação Matemática

Seja f é uma função não linear, tal que, $f : \mathbb{R}^n \rightarrow \mathbb{R}$.

Pretende-se determinar o ponto $x^* \in \mathbb{R}^n$, tal que

$$\begin{array}{ll} \min & f(x) \\ \text{s.a} & c_i(x) = 0, \quad i \in I \\ & c_i(x) \geq 0, \quad i \in D \end{array} \quad (PO)$$

Exemplos

Exemplo 1

$$\begin{array}{ll} \min & (x - 2)^2 \\ x \in \mathbf{R} & x - 3 \geq 0 \end{array}$$

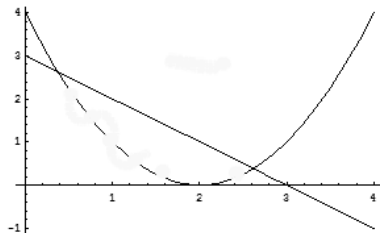


Figura : Optimização com restrições

Definições

Diz-se:

- **Restrição activa** - verifica-se a igualdade em x_k , isto é, $c_i(x_k) = 0$.
- **Restrição inactiva** - verifica-se a desigualdade em x_k , isto é, $c_i(x_k) > 0$.

Região admissível - todos os pontos que satisfazem todas as restrições do problema, isto é,

$$\Omega = \{x \in \mathbf{R}^n, c_i(x) = 0, i \in I, c_i(x) \geq 0, i \in D\}.$$

Exemplos

Exemplo 2

$$\begin{aligned} \min \quad & -x_1x_2 \\ & -x_1 - x_2^2 + 1 \geq 0 \\ & x_1 + x_2 \geq 0 \end{aligned}$$

- Qual a região admissível?
- Indique um ponto com alguma restrição activa.
- Indique um ponto com alguma restrição inactiva.

Função Lagrangeana

Seja $x \in \mathbb{R}^n$.

Função Langrangeana

$$L(x, \lambda) = f(x) - \sum_{i \in I \cup D} \lambda_i c_i(x)$$

onde:

- $f(x)$ – função objectivo,
- λ_i – multiplicador Lagrangeano associado à restrição i ,
- $c_i(x)$ – restrição i .

Conjunto Activo

Seja $\bar{x} \in \mathbb{R}^n$.

Conjunto Activo

$$A(\bar{x}) = I \cup \{i \in D : c_i(\bar{x}) = 0\}$$

Conjunto de índices para os quais as restrições verificam a igualdade quando avaliados no ponto $\bar{x}$.

Definições

Qualificação de restrição independência linear

Dado um ponto $\bar{x}$, diz-se que a **qualificação de restrição independência linear** (QRIL) é verificada se o conjunto de vectores

$$\{\nabla c_i(\bar{x}), i \in A(\bar{x})\}$$

é linearmente independente.

Esta condição também é designada por Condição de Regularidade.

Ao ponto $\bar{x}$ designa-se por ponto regular.

Condições necessárias e suficientes de 1ª ordem

Seja x^* uma solução local do problema (PO) e a QRLI é verificada em x^* . Então existe um vector de multiplicadores de Lagrange, λ^* , com componentes λ_i^* , para $i \in I \cup D$, tal que as seguintes condições são satisfeitas em (x^*, λ^*)

$$\begin{aligned}\nabla_x L(x^*, \lambda^*) &= 0, \\ c_i(x^*) &= 0, \text{ para } i \in I, \\ c_i(x^*) &\geq 0, \text{ para } i \in D, \\ \lambda_i^* &\geq 0, \text{ para } i \in D, \\ \lambda_i^* c_i(x^*) &= 0, \text{ para } i \in I \cup D,\end{aligned}$$

Também designadas por condições KKT.

Ao ponto (x^*, λ^*) é o ponto KKT.

Condições suficientes de 2ª ordem

Seja (x^*, λ^*) um ponto KKT e se

$$s^T \nabla_{xx}^2 L(x^*) s > 0, \text{ para } s \in F_2(\lambda^*), s \neq 0.$$

Então x^* é minimizante local forte de $f(x)$ na região admissível.

Com

$$s \in F_2(\lambda^*)$$



$$\begin{cases} \nabla c_i(x^*)^T s = 0, & \text{para todo } i \in I, \\ \nabla c_i(x^*)^T s = 0, & \text{para todo } i \in A(x^*) \cap D \text{ com } \lambda_i^* > 0, \\ \nabla c_i(x^*)^T s \geq 0, & \text{para todo } i \in A(x^*) \cap D \text{ com } \lambda_i^* = 0, \end{cases}$$

Métodos iterativos

Os métodos iterativos para a resolução de problemas de optimização com restrições:

- Método de penalidade,
- Método da programação quadrática sequencial,
- Método de pontos interiores,
- ...

Método de penalidade

Característica

O método de penalidade procura a aproximação à solução do problema de optimização com restrições resolvendo uma sucessão de subproblemas sem restrições.

A função objectivo dos subproblemas é constituída por:

- função objectivo do problema original, e
- um termo associado a cada restrição - que terá valor positivo se a restrição for violada e o valor nulo caso contrário.

Método de penalidade

Alguns exemplos de técnicas do método de penalidade:

- Método de penalidade quadrática
- Método de barreira
- Método de penalidade exponencial
-

Método de penalidade quadrática

Função de penalidade quadrática, $Q(x, \mu)$, definida por:

$$Q(x, \mu) = f(x) + \frac{1}{2\mu} \sum_{i \in I} c_i^2(x) + \frac{1}{2\mu} \sum_{i \in D} (\max\{-c_i(x), 0\})^2$$

onde μ é um parametro de penalidade que decresce para o valor zero.

Método de penalidade quadrática

Seja $k = 1$ e considere uma aproximação inicial x_0 , $\mu_0 > 0$ e $\epsilon_0 > 0$.

Método de penalidade quadrática

- Passo 1: Resolver o subproblema

$$\min Q_k(x) \equiv Q(x, \mu_k).$$

- Passo 2: Actualizar o parâmetro de penalidade $\mu_{k+1} \in (0, \mu_k)$ e o parâmetro ϵ_{k+1} .
- Passo 3: Se não for satisfeito o critério de paragem ir para o Passo 1 com $k = k + 1$.

Método de penalidade quadrática

Passo 1

Usar um método descrito no Capítulo 2.

Terminar o processo iterativo quando $\|x_{k+1} - x_k\| \leq \epsilon_k$

Passo 2

Actualizar $\mu_{k+1} = c\mu_k$, com $0 < c < 1$ ($c = 0.7$ ou $c = 0.1$).

Actualizar $\epsilon_{k+1} = s\epsilon_k$, com $0 < s < 1$.

Método de penalidade quadrática

Propriedades

Se x_k for o minimizante global exacto de $Q(x, \mu_k)$ e a sucessão $\{\mu_k\}_{k \geq 1}$ decresce para zero. Então qualquer ponto limite x^* da sucessão $\{x_k\}_{k \geq 1}$ é solução do problema (PO).

Método de penalidade l_1

Função de penalidade l_1 , $\phi_1(x, \mu)$, definida por:

$$\phi_1(x, \mu) = f(x) + \frac{1}{\mu} \sum_{i \in I} |c_i(x)| + \frac{1}{\mu} \sum_{i \in D} \max\{-c_i(x), 0\}$$

onde μ é um parametro de penalidade que decresce para o valor zero.

Método de penalidade l_1

Seja $k = 1$ e considere uma aproximação inicial x_0 , $\mu_0 > 0$ e $\epsilon_0 > 0$.

Método de penalidade l_1

- Passo 1: Resolver o subproblema

$$\min \phi_1^k(x) \equiv \phi_1(x, \mu_k).$$

- Passo 2: Actualizar o parâmetro de penalidade $\mu_{k+1} \in (0, \mu_k)$ e o parâmetro ϵ_{k+1} .
- Passo 3: Se não for satisfeito o critério de paragem ir para o Passo 1 com $k = k + 1$.

Método de penalidade l_1

Propriedades

ϕ_1 é uma função de **penalidade exacta** - existe um valor μ para o qual a solução do problema (PO) é a solução de

$$\min \phi_1(x, \mu)$$

Propriedades

ϕ_1 é uma função não suave!!!!

Método da Lagrangeana aumentada

Este método também é designado por Método dos Multiplicadores.

Função de penalidade L_A , $L_A(x, \lambda, \mu)$, definida por:

$$L_A(x, \lambda, \mu) = f(x) - \sum_{i \in I} \lambda_i c_i(x) + \frac{1}{2\mu} \sum_{i \in I} c_i^2(x)$$

onde λ é o multiplicador de Lagrange e μ é um parâmetro de penalidade que decresce para o valor zero.

Método da Lagrangeana aumentada

Este método também é designado por Método dos Multiplicadores.

Função de penalidade L_A , $L_A(x, \lambda, \mu)$, definida por:

$$L_A(x, \lambda, \mu) = f(x) - \sum_{i \in I} \lambda_i c_i(x) + \frac{1}{2\mu} \sum_{i \in I} c_i^2(x)$$

onde λ é o multiplicador de Lagrange e μ é um parâmetro de penalidade que decresce para o valor zero.

- Só restrições de igualdades!!!!

Método da Lagrangeana aumentada

Este método também é designado por Método dos Multiplicadores.

Função de penalidade L_A , $L_A(x, \lambda, \mu)$, definida por:

$$L_A(x, \lambda, \mu) = f(x) - \sum_{i \in I} \lambda_i c_i(x) + \frac{1}{2\mu} \sum_{i \in I} c_i^2(x)$$

onde λ é o multiplicador de Lagrange e μ é um parâmetro de penalidade que decresce para o valor zero.

- **Só restrições de igualdades!!!!**
- Se houver, no problema original, restrições de desigualdade é necessário transformá-las em restrições de igualdade!

Método da Lagrangeana aumentada

Seja $k = 1$ e considere uma aproximação inicial x_0 , λ^0 , $\mu_0 > 0$ e $\epsilon_0 > 0$.

Método de penalidade L_A

- Passo 1: Resolver o subproblema

$$\min L_A^k(x) \equiv L_A(x, \lambda^k, \mu_k).$$

- Passo 2: Actualizar o parâmetro de penalidade $\mu_{k+1} \in (0, \mu_k)$, o multiplicador de Lagrange λ^k e o parâmetro ϵ_{k+1} .
- Passo 3: Se não for satisfeito o critério de paragem ir para o Passo 1 com $k = k + 1$.

Método da Lagrangeana aumentada

Passo 1

Usar um método descrito no Capítulo 2.

Terminar o processo iterativo quando $\|\nabla L_A^k(x_{k+1})\| \leq \epsilon_k$

Passo 2

Actualizar $\lambda_i^{k+1} = \lambda_i^k - c_i(x_{k+1})/\mu_k$, para $i \in I$.

Actualizar $\mu_{k+1} = c\mu_k$, com $0 < c < 1$ ($c = 0.7$ ou $c = 0.1$).

Actualizar $\epsilon_{k+1} = \min\{\eta_k, \gamma_k \|c(x_{k+1})\|\}$.

As sucessões $\{\eta_k\}_{k \geq 1}$ e $\{\gamma_k\}_{k \geq 1}$ decrescem para zero.

Método da Lagrangeana aumentada

Propriedades

Seja x^* um minimizante local do problema (PO) onde a QRLI e as condições suficientes de segunda ordem são satisfeitas. Então existe $\bar{\mu}$, tal que, qualquer que seja $\mu \in (0, \bar{\mu})$, x^* é um minimizante local isolado de $L_A(x, \lambda^*, \mu)$.

Exercícios

Considere o seguinte problema

$$\begin{aligned} \min \quad & x_1 + x_2 \\ & x_1^2 + x_2^2 - 2 = 0 \\ & x_1 + x_2 \geq 0 \end{aligned}$$

- Resolva-o usando o método de penalidade quadrática.
- Resolva-o usando o método de penalidade l_1 .
- Resolva-o usando o método da Lagrangeana aumentada.

Considere as aproximações iniciais $x_0 = (0, 0)$ e $y_0 = (1, -2)$.

Método da programação quadrática sequencial

$$\begin{array}{ll}\min & f(x) \\ \text{s.a} & c_i(x) = 0, \quad i \in I \\ & c_i(x) \geq 0, \quad i \in D\end{array}$$

O método da programação quadrática sequencial (SQP) aproxima localmente o problema inicial!!

Método da programação quadrática sequencial

O método SQP aproxima localmente o problema inicial por um problema mais simples de programação quadrática.

- a função objectivo $f(x)$ - é aproximada por um modelo quadrático,
- a função restrição $c_i(x)$ - é aproximada por um modelo linear.

Método da programação quadrática sequencial

A série de Taylor de $g(x)$, em torno do ponto $\bar{x}$, em $\bar{x} + p$, é dada por:

$$g(\bar{x} + p) = g(\bar{x}) + p^T \nabla g(\bar{x}) + \frac{1}{2} p^T \nabla^2 g(\bar{x}) p + \dots$$

Seja x_k uma aproximação ao problema inicial.

- modelo quadrático de $f(x)$

$$f(x_k) + \nabla^T f(x_k) p + \frac{1}{2} p^T \nabla^2 f(x_k) p,$$

- modelo linear $c_i(x)$

$$\nabla^T c_i(x_k) p + c_i(x_k).$$

Método da programação quadrática sequencial

Seja x_k uma aproximação ao problema inicial.

$$\begin{aligned} \min \quad & f(x_k) + \nabla^T f(x_k)p + \frac{1}{2}p^T \nabla^2 f(x_k)p \\ p \in \mathbb{R}^n, \quad & \nabla^T c_i(x_k)p + c_i(x_k) = 0, \quad i \in I \\ & \nabla^T c_i(x_k)p + c_i(x_k) \geq 0, \quad i \in D \end{aligned}$$

ou simplesmente

$$\begin{aligned} \min \quad & \nabla^T f(x_k)p + \frac{1}{2}p^T \nabla^2 f(x_k)p \\ p \in \mathbb{R}^n, \quad & \nabla^T c_i(x_k)p + c_i(x_k) = 0, \quad i \in I \\ & \nabla^T c_i(x_k)p + c_i(x_k) \geq 0, \quad i \in D \end{aligned}$$

A solução é o vector p_k - **a direcção de procura.**

Método da programação quadrática sequencial

Esta aproximação é válida numa pequena vizinhança de x_k .

A nova aproximação à solução será $x_{k+1} = x_k + p_k$.

O processo deverá ser repetido até ser satisfeito o critério de paragem (o ponto x_{k+1} satisfaz as condições KKT.).

Método da programação quadrática sequencial

Em vez de $\nabla^2 f(x_k)$ é mais vantajoso usar a matriz Hessiana da Lagrangiana $\nabla_{xx}^2 L(x_k, \lambda_k)$!!!

Relembrar, a função Lagrangeana:

$$L(x, \lambda) = f(x) - \sum_{i \in I \cup D} \lambda_i c_i(x)$$

Resolução do problema quadrático

Através das condições KKT do problema!!!

Técnicas de globalização

O método SQP é local.

Logo precisamos de uma técnica de globalização!!!!

Sugestão

Critério de Armijo + Função Mérito (M)

Condição de Armijo

Dados x_k e p_k , calcular $\alpha_k \in (0, 1]$ tal que satisfaz a seguinte condição

Condição de Armijo

$$M(x_k + \alpha_k p_k) \leq M(x_k) + \mu \alpha_k \nabla M(x_k)^T p_k$$

Função mérito

Existem várias funções mérito, que devem depender da função objectivo e das funções das restrições:

- Função Lagrangeana aumentada
- ...

Exercício

Considere o seguinte problema

$$\begin{aligned} \min \quad & e^{x_1} + x_2 \\ & x_1^2 + x_2^2 - 2 = 0 \\ & x_1 + x_2 \geq 0 \end{aligned}$$

Resolver com o método SQP considerando $x_0 = (0, 0)$.

Função: fmincon

A função `fmincon` é baseada no SQP.

$$\begin{array}{ll}
 \min & f(x) \\
 \text{s.a} & Ax \leq b \quad \text{(lineares de desigualdades)} \\
 & Aeqx = beq \quad \text{(lineares de igualdades)} \\
 & lb \leq x \leq ub \quad \text{(limites simples)} \\
 & c(x) \leq 0 \quad \text{(não lineares de desigualdades)} \\
 & ceq = 0 \quad \text{(não lineares de igualdades)}
 \end{array}$$

Função: fmincon

Sintaxe:

`x = fmincon(fun,x0,A,b)`

`x = fmincon(fun,x0,A,b,Aeq,beq)`

`x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub)`

`x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)`

`x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)`

`[x,fval] = fmincon(...)`

`[x,fval,exitflag] = fmincon(...)`

`[x,fval,exitflag,output] = fmincon(...)`

`[x,fval,exitflag,output,lambda] = fmincon(...)`

`[x,fval,exitflag,output,lambda,grad] = fmincon(...)`

`[x,fval,exitflag,output,lambda,grad,hessian] = fmincon(...)`

Função: fmincon

Variáveis de entrada:

- fun - M-file com a função objectivo
- x0 - ponto inicial
- A - Matriz dos coeficientes das restrição lineares de desigualdade
- b - Vector independente das restrições lineares de desigualdade
- Aeq - Matriz dos coeficientes das restrição lineares de igualdade
- beq - Vector independente das restrições lineares de igualdade

Função: fmincon

Variáveis de entrada (cont.):

- `lb, ub` - Vectors dos limites inferiores e superiores de x
- `nonlcon` - M-file com informação das restrições não lineares (igualdade e desigualdade)
- `options` - opções criadas com `optimset`

Função: fmincon

Variáveis de saída:

Idênticas à função `fminunc`.

Exercícios

Considere o seguinte problema

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & f(x) = x_1^2 + x_2^2 - (1 - x_1 x_2)^2 \\ & |x_1| + |x_2| \leq 1 \\ & (x_1 - 1)^2 + x_2^2 = 1 \\ & x_1^2 + x_2^2 \leq 0.5 \\ & x_2 \leq 0.25 \end{aligned}$$

Use a função `fmincon` com as aproximações iniciais $x_0 = (1, 0.7)$ e $y_0 = (1, 2)$.

Referencias



E. Fernandes, I. Vaz, T. Monteiro, I. Santo. Curso de optimização com aplicações à biomedicina. NORG. 2008.



J. Nocedal e S. Wright. Numerical Optimization. *Springer Series in Operations Research*. Springer. 1999.



S. G. Nash e A. Sofer. Linear and Nonlinear Programming. *Industrial Engineering Series*. McGraw-Hill International Editions. 1996.



M. S. Bazaraa, H. D. Sherali e C. M. Shetty. Nonlinear Programming Theory and Algorithms. John Wiley and Sons. 1993.



MatLab, website: www.mathworks.com

Capítulo 4: Equações diferenciais ordinárias

Ana I. Pereira[‡]

[‡] Instituto Politécnico de Bragança, Portugal, apereira@ipb.pt

Matemática Aplicada

Problema

Neste problema pretende-se determinar $y(t)$ tal que

$$\frac{dy(t)}{dt} = f(t, y(t)) \quad (1)$$

A equação (1) é designada por **equação diferencial ordinária**.

A variável t designa-se por variável independente.

A variável y designa-se por variável dependente.

Exemplo

$$\frac{dy(t)}{dt} = e^t$$

Equação diferencial ordinária

Equação diferencial ordinária

Designamos por equação diferencial ordinária à equação que contém a derivada de uma função desconhecida. A função desconhecida apenas depende de uma variável.

Ordem da equação diferencial ordinária

Designamos por ordem da equação diferencial ordinária à ordem mais elevada da derivada presente na equação.

Exemplos

- $y'' + y' - 1 = 0$
- $y' - e^t = 2$
- $y''' + y'' = 1$

Problema de valor inicial

Designa-se por problema de valor inicial (PVI) ao problema

$$\frac{dy(t)}{dt} = f(t, y(t))$$

onde $y(t_0) = y_0$ e $a \leq t \leq b$.

Exemplo

$$\frac{dy(t)}{dt} = e^t$$

com $y(0) = 2$ e $0 \leq t \leq 10$.

Métodos Numéricos para PVI

A solução numérica para o problema de valor inicial é uma sucessão de valores para as variáveis independentes

$$t_0, t_1, \dots, t_n$$

e para a variável dependente

$$y_0, y_1, \dots, y_n.$$

onde y_i representa uma aproximação para $y(t_i)$:

$$y_i \approx y(t_i), \text{ para } i = 0, 1, \dots, n.$$

Métodos Numéricos para PVI

Consideremos $t_0 = a$ e $t_n = b$.

O comprimento do passo é dado por $h_i = t_{i+1} - t_i$.

Notação:

$$\dot{y} = \frac{dy(t)}{dt} = y'(t)$$

Método de Euler

Consideremos um comprimento do passo fixo, h .

A expansão em série de Taylor de $y(t + h)$ é dada por

$$y(t+h) = y(t) + hy'(t) + \frac{h^2}{2}y''(t) + \dots$$

Como $y'(t) = f(t, y(t))$, obtenemos

$$y(t+h) = y(t) + hf(t,y(t)) + \frac{h^2}{2} \frac{df(t,y(t))}{dt} + \dots$$

Método de Euler

O método de Euler é o método mais simples para resolver PVI. Consideremos um comprimento do passo fixo, h .

São dados t_0 e y_0 .

A sucessão de aproximações à solução é dada por:

$$\begin{aligned}y_{i+1} &= y_i + hf_i \\ t_{i+1} &= t_i + h\end{aligned}$$

para $i = 0, 1, \dots, n - 1$. E $f_i = f(t_i, y_i)$.

Método de Euler

Exemplo 1

Use o método de Euler para determinar a solução do PVI

$$\frac{dy(t)}{dt} = e^t$$

com $y(0) = 2$ e $0 \leq t \leq 10$.

Use $h = 1.0$ e $h = 0.5$.

Exemplo 2

Use o método de Euler para determinar a solução do PVI

$$y' = y$$

com $y(0) = 1$ e $0 \leq t \leq 5$.

Use $h = 0.5$ e $h = 0.25$.

Método de Euler modificado

Existem diversas variantes do Método de Euler.

A ideia base é

$$y_{i+1} = y_i + h\bar{f}$$

onde:

- $\bar{f}$ é uma aproximação da derivada de y entre os pontos t_i e t_{i+1} ;
- h representa o comprimento do passo.

Método de Euler modificado

Uma variante é a sucessão de aproximações à solução ser dada por:

$$\begin{aligned}t_{i+1} &= t_i + h \\ y_{i+1} &= y_i + hf(t_i, y_i) \\ y_{i+1} &= y_i + hf(t_{i+1}, y_{i+1})\end{aligned}$$

para $i = 0, 1, \dots, n - 2$.

y_{i+1} é calculado inicialmente pelo método de Euler.

Método de Euler modificado

Exemplo 1

Use o método de Euler modificado para determinar a solução do PVI

$$\frac{dy(t)}{dt} = e^t$$

com $y(0) = 2$ e $0 \leq t \leq 10$.

Use $h = 1.0$ e $h = 0.5$.

Exemplo 2

Use o método de Euler modificado para determinar a solução do PVI

$$y' = y$$

com $y(0) = 1$ e $0 \leq t \leq 5$.

Use $h = 0.5$ e $h = 0.25$.

Método de Heun

O método de Heun tem como base o método de Euler.

A ideia base é

$$y_{i+1} = y_i + \frac{h}{2}(f_i + f_{i+1}) \quad (2)$$

onde $f_{i+1} = f(t_{i+1}, y_{i+1})$ depende do valor y_{i+1} , logo não podemos resolver a expressão (2) directamente.

Método de Heun

Neste método, a sucessão de aproximações à solução é dada por:

$$\begin{aligned}f_i &= f(t_i, y_i) \\ y_{i+1} &= y_i + hf_i \\ t_{i+1} &= t_i + h \\ f_{i+1} &= f(t_{i+1}, y_{i+1}) \\ y_{i+1} &= y_i + \frac{h}{2}(f_i + f_{i+1})\end{aligned}$$

para $i = 0, 1, \dots, n - 1$.

Para iniciar o processo necessitamos de h e y_0 .

Método de Heun

Exemplo 1

Use o método de Heun para determinar a solução do PVI

$$\frac{dy(t)}{dt} = e^t$$

com $y(0) = 2$ e $0 \leq t \leq 10$.

Use $h = 1.0$ e $h = 0.5$.

Exemplo 2

Use o método de Heun para determinar a solução do PVI

$$y' = y$$

com $y(0) = 1$ e $0 \leq t \leq 5$.

Use $h = 0.5$ e $h = 0.25$.

Método de Runge-Kutta

O método de Runge-Kutta usa uma média ponderada.
A ideia base do método de Runge-Kutta de ordem m é dado por

$$\begin{aligned} s_j &= f(t_i + \alpha_j h, y_i + h \sum_{k=1}^{j-1} \beta_{j,k} s_k) \text{ para } j = 1, \dots, m \\ y_{i+1} &= y_i + h \sum_{j=1}^m \gamma_j s_j \end{aligned} \quad (3)$$

Método de Runge-Kutta de ordem 4

No método de Runge-Kutta de ordem 4, a sucessão de aproximações à solução é dada por:

$$\begin{aligned} s_1 &= f(t_i, y_i) \\ s_2 &= f\left(t_i + \frac{h}{2}, y_i + \frac{h}{2}s_1\right) \\ s_3 &= f\left(t_i + \frac{h}{2}, y_i + \frac{h}{2}s_2\right) \\ s_4 &= f(t_i + h, y_i + hs_3) \\ y_{i+1} &= y_i + \frac{h}{6}(s_1 + 2s_2 + 2s_3 + s_4) \\ t_{i+1} &= t_i + h \end{aligned}$$

para $i = 0, 1, \dots, n - 1$.

Para iniciar o processo necessitamos de h e y_0 .

Método de Runge-Kutta

Exemplo 1

Use o método de Runge-Kutta para determinar a solução do PVI

$$\frac{dy(t)}{dt} = e^t$$

com $y(0) = 2$ e $0 \leq t \leq 10$.

Use $h = 1.0$ e $h = 0.5$.

Exemplo 2

Use o método de Runge-Kutta para determinar a solução do PVI

$$y' = y$$

com $y(0) = 1$ e $0 < t < 5$.

Matlab - Resolução exacta

`dsolve`

`dsolve` - Symbolic solution of ordinary differential equations.

`dsolve('eq1','eq2',...,'cond1','cond2',...,'v')`

- `eq1` - equações,
- `cond1` - condições,
- `v` - variável independente.

Matlab - Resolução numérica

ode23, ode45, ode113, ode15s, ode23s, ode23t, ode23tb - Solve initial value problems for ordinary differential equations.

```
[T,Y] = solver(odefun,tspan,y0)
[T,Y] = solver(odefun,tspan,y0,options)
[T,Y,TE,YE,IE] =
solver(odefun,tspan,y0,options)
sol = solver(odefun,[t0 tf],y0...)
```

Matlab - Resolução numérica

Variáveis de entrada

- `odefun` - function f .
- `tspan` - vector representante do intervalo da variável independente.
- `y0` - vector das condições iniciais.
- `options` - opções.

Matlab - Resolução numérica

Variáveis de saída

- T - Column vector of time points.
- Y - Solution array. Each row in Y corresponds to the solution at a time returned in the corresponding row of T.
- TE - The time at which an event occurs.
- YE - The solution at the time of the event.
- IE - The index i of the event function that vanishes.
- sol - Structure to evaluate the solution.

Referencias



J. Nocedal e S. Wright. Numerical Optimization. *Springer Series in Operations Research*. Springer. 1999.



S. G. Nash e A. Sofer. Linear and Nonlinear Programming. *Industrial Engineering Series*. McGraw-Hill International Editions. 1996.



M. S. Bazaraa, H. D. Sherali e C. M. Shetty. Nonlinear Programming Theory and Algorithms. John Wiley and Sons. 1993.

Capítulo 5: Equações diferenciais ordinárias de ordem superior

Ana I. Pereira[‡]

[‡] Instituto Politécnico de Bragança, Portugal, apereira@ipb.pt

Matemática Aplicada

Problema

Neste problema pretende-se determinar $y(t)$ tal que

$$\frac{d^n y(t)}{dt^n} = f\left(t, y(t), \frac{dy(t)}{dt}, \dots, \frac{d^{n-1}y(t)}{dt^{n-1}}\right) \quad (1)$$

A equação (3) é designada por **equação diferencial ordinária de orden n** .

Exemplo

$$\frac{d^2 y(t)}{dt^2} + \frac{dy(t)}{dt} = e^t$$

Problema

Considere o problema pretende-se determinar $y(t)$ tal que

$$\frac{d^n y(t)}{dt^n} = f\left(t, y(t), \frac{dy(t)}{dt}, \dots, \frac{d^{n-1}y(t)}{dt^{n-1}}\right) \quad (2)$$

$$\text{E } y(t_0) = y_0, \frac{dy(t_0)}{dt} = y_0^{(1)}, \dots, \frac{d^{n-1}y(t_0)}{dt^{n-1}} = y_0^{(n-1)}$$

Problema

Transforma-se o problema (3) num sistema definido

Exemplo

$$z = \begin{bmatrix} z_1 \\ z_2 \\ \dots \\ z_n \end{bmatrix} = \begin{bmatrix} y \\ \frac{dy(t)}{dt} \\ \dots \\ \frac{d^{n-1}y(t)}{dt^{n-1}} \end{bmatrix}$$

Problema

Ora

$$z' = \begin{bmatrix} z'_1 \\ z'_2 \\ \dots \\ z'_n \end{bmatrix} = \begin{bmatrix} y' \\ y'' \\ \dots \\ y^{(n)} \end{bmatrix} = \begin{bmatrix} z_2 \\ z_3 \\ \dots \\ f(t, y(t), y', \dots, y^{(n-1)}) \end{bmatrix}$$

Exemplo

exemplo

Considere o problema pretende-se determinar $y(t)$ tal que

$$\frac{d^2y(t)}{dt^2} + \frac{dy(t)}{dt} = yt \quad (3)$$

E $y(0) = 0$, $\frac{dy(0)}{dt} = 1$ e $t \in [0, 1]$.

Transforma-se o problema (3) num sistema definido

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} y \\ \frac{dy(t)}{dt} \end{bmatrix}$$

Exemplo

Assim,

$$z' = \begin{bmatrix} z_1' \\ z_2' \end{bmatrix} = \begin{bmatrix} z_2 \\ z_1 t - z_2 \end{bmatrix}$$

Consideremos $t_0 = 0$ e $t_n = 1$.

O comprimento do passo é dado por $h_i = 0.25$.

Seja $z^{(0)} = (0, 1)$

Método de Euler

O método de Euler é o método mais simples para resolver PVI.

Consideremos um comprimento do passo fixo, h .

São dados t_0 e $z^{(0)}$.

A sucessão de aproximações à solução é dada por:

$$\begin{aligned}z^{(i+1)} &= z^{(i)} + hF_i \\t_{i+1} &= t_i + h\end{aligned}$$

para $i = 0, 1, \dots, n - 1$.

Método de Euler

Exemplo 1

Use o método de Euler para determinar a solução do PVI

$$\frac{d^2y(t)}{dt^2} + \frac{dy(t)}{dt} = yt \quad (4)$$

com $y(0) = 0$, $y'(0) = 1$ e $0 \leq t \leq 1$.

Use $h = 0.25$ e $h = 0.5$.

Método de Euler

Exemplo 2

Use o método de Euler para determinar a solução do PVI

$$y'' + 3y' = y + \cos(t)$$

com $y(0) = 1$, $y'(0) = 2$ e $0 \leq t \leq 2$.

Use $h = 0.5$ e $h = 0.25$.

Método de Euler modificado

Devem ser feitas as alterações convenientes para os restantes métodos.

Matlab - Resolução exacta

`dsolve`

`dsolve` - Symbolic solution of ordinary differential equations.

`dsolve('eq1','eq2',...,'cond1','cond2',...,'v')`

- `eq1, eq2,...` - equações,
- `cond1, cond2,...` - condições,
- `v` - variável independente.

Matlab - Resolução numérica

ode23, ode45, ode113, ode15s, ode23s, ode23t, ode23tb - Solve initial value problems for ordinary differential equations.

```
[T,Y] = solver(odefun,tspan,y0)
[T,Y] = solver(odefun,tspan,y0,options)
[T,Y,TE,YE,IE] =
solver(odefun,tspan,y0,options)
sol = solver(odefun,[t0 tf],y0...)
```

Matlab - Resolução numérica

Variáveis de entrada

- `odefun` - function f .
- `tspan` - vector representante do intervalo da variável independente.
- `y0` - vector das condições iniciais.
- `options` - opções.

Matlab - Resolução numérica

Variáveis de saída

- T - Column vector of time points.
- Y - Solution array. Each row in Y corresponds to the solution at a time returned in the corresponding row of T.
- TE - The time at which an event occurs.
- YE - The solution at the time of the event.
- IE - The index i of the event function that vanishes.
- sol - Structure to evaluate the solution.

Referencias



J. Nocedal e S. Wright. Numerical Optimization. *Springer Series in Operations Research*. Springer. 1999.



S. G. Nash e A. Sofer. Linear and Nonlinear Programming. *Industrial Engineering Series*. McGraw-Hill International Editions. 1996.



M. S. Bazaraa, H. D. Sherali e C. M. Shetty. Nonlinear Programming Theory and Algorithms. John Wiley and Sons. 1993.

Capítulo 6: Equações diferenciais parciais

Ana I. Pereira[‡]

[‡] Instituto Politécnico de Bragança, Portugal, apereira@ipb.pt

Matemática Aplicada, Mestrados em Tecnologia Biomédica,
Engenharia Química e Engenharia Industrial

Introdução

A equação (1) é uma equação geral com derivadas parciais de segunda ordem.

Neste capítulo pretende-se determinar $u(x, y)$ tal que

$$A(x, y) \frac{\partial^2 u}{\partial x^2} + B(x, y) \frac{\partial^2 u}{\partial x \partial y} + C(x, y) \frac{\partial^2 u}{\partial y^2} = f \left(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \right) \quad (1)$$

com as seguintes condições

$$\begin{aligned} u(x, y_0) &= b_{y_0}(x), & u(x, y_f) &= b_{y_f}(x) \\ u(x_0, y) &= b_{x_0}(y), & u(x_f, y) &= b_{x_f}(y) \end{aligned}$$

Exemplo

Determinar $u(x, y)$ tal que

$$u_{xx} - xu_{xy} + 4u_{yy} = 0$$

com $0 \leq x \leq 4$ e $0 \leq y \leq 4$ com as seguintes condições de fronteira

$$\begin{aligned} u(x, 0) &= e^x, & u(x, 4) &= e^x \cos(4) \\ u(0, y) &= \cos(y), & u(4, y) &= e^4 \cos(y) \end{aligned}$$

As EDP podem ser classificadas em três tipos:

Exemplo

- EDP Elíptica: se $B^2 - 4AC < 0$.
- EDP Parabólica: se $B^2 - 4AC = 0$.
- EDP Hiperbólica: se $B^2 - 4AC > 0$.

EDP Elíptica

A equação de Poisson é um exemplo de uma equação diferencial parcial elíptica,

$$u_{xx} + u_{yy} = f(x, y)$$

com $a \leq x \leq b$ e $c \leq y \leq d$, e satisfazendo as seguintes condições

$$\begin{aligned} u(x, 0) &= g_1(x), & u(x, 4) &= g_2(x) \\ u(0, y) &= g_3(y), & u(4, y) &= g_4(y) \end{aligned}$$

EDP Elíptica - Método das diferenças divididas

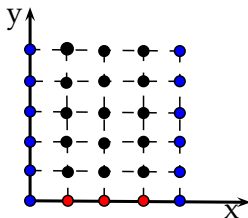
Método das diferenças divididas

- Discretizamos o espaço $[a, b] \times [c, d]$ por $x_0, x_1, \dots, x_{N_x}$ e $y_0, y_1, \dots, y_{N_y}$ com pontos igualmente espaçados. O espaçamento é dado por $h_x = x_{i+1} - x_i$ e $h_y = y_{j+1} - y_j$.
- Substituir as derivadas por aproximações de diferenças divididas.
- Resolver o sistema.

Descobrir $u(x_i, y_j)$, para $i = 0, \dots, N_x$ e $j = 0, \dots, N_y$, usando as diferenças divididas!!!

Grelha

Um exemplo de grelha é dada por:



EDP Elíptica

Substituir as derivadas de segunda ordem por uma aproximação de diferenças finitas

$$u_{xx}|_{x_i, y_j} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2}$$

$$u_{yy}|_{x_i, y_j} \approx \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_y^2}$$

com $u_{i,j} = u(x_i, x_j)$ para $1 \leq i \leq N_x - 1$ e $1 \leq j \leq N_y - 1$

EDP Elíptica

Substituímos na equação e obtemos

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_y^2} = f(x,y)$$

obtemos um sistema de equações, que é necessário resolver!!

EDP Elíptica - Exemplo

Exemplo

Resolva a seguinte equação

$$u_{xx} + u_{yy} = 0$$

para $0 \leq x \leq 1$ e $0 \leq y \leq 1$, sabendo que

$$u(0, y) = 0, u(x, 0) = 0, u(x, 1) = 200x \text{ e } u(1, y) = 200y.$$

Considere $h_x = 0.5$ e $h_y = 0.5$.

EDP Elíptica - Exemplo

Substituímos na equação e obtemos

$$\frac{u_{2,1} - 2u_{1,1} + u_{0,1}}{0.5^2} + \frac{u_{1,2} - 2u_{1,1} + u_{1,0}}{0.5^2} = 0$$

Como conhecemos $u_{0,0}$, $u_{1,0}$, $u_{2,0}$, $u_{0,1}$, $u_{0,2}$, $u_{2,1}$, $u_{2,2}$ e $u_{1,2}$.
Basta descobrir $u_{1,1}$!!!

Uma equação. Uma variável.

EDP Parabólica

A equação do calor é um exemplo de uma equação diferencial parabólica

Equação do Calor

$$\frac{\partial u}{\partial y} = \alpha^2 \frac{\partial^2 u}{\partial x^2}$$

sujeita às seguintes condições

$$u(0, y) = u(l, y) = 0, \quad y > 0$$

e

$$u(x, 0) = f(x), \quad 0 \leq x \leq l$$

EDP Parabólica - Método das diferenças finitas

Discretizamos o espaço $x_0, x_1, \dots, x_{N_x}$ e $y_0, y_1, \dots, y_{N_y}$ com pontos igualmente espaçados. O espaçamento é dado por $h_x = x_{i+1} - x_i$ e $h_y = y_{j+1} - y_j$.

Descobrir $u(x_i, y_j)$, para $i = 0, \dots, N_x$ e $j = 0, \dots, N_y$, usando o método das diferenças divididas!!!

EDP Parabólica

Substituir as derivadas por uma aproximação de diferenças finitas

$$u_y|_{x_i, y_j} \approx \frac{u_{i,j+1} - u_{i,j}}{h_y}$$

e

$$u_{xx}|_{x_i, y_j} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2}$$

com $u_{i,j} = u(x_i, x_j)$ para $1 \leq i \leq N_x - 1$ e $0 \leq j \leq N_y - 1$

EDP Parabólica

Substituímos na equação e obtemos

$$\frac{u_{i,j+1} - u_{i,j}}{h_y} - \alpha^2 \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2} = 0$$

obtemos um sistema de equações, que é necessário resolver!!

EDP Parabólica - Exemplo

Considere a equação do calor

$$\frac{\partial u}{\partial y} - \frac{\partial^2 u}{\partial x^2} = 0$$

sujeita às seguintes condições

$$u(0, y) = u(1, y) = 0, \quad y > 0$$

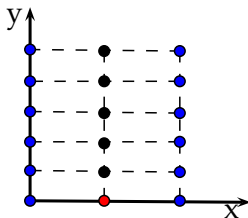
e

$$u(x, 0) = \sin(\pi x), \quad 0 \leq x \leq 1$$

Resolva o problema considerando $h_x = 0.5$ e $h_y = 0.2$.

EDP Parabólica - Exemplo

A grelha é dada por:



EDP Parabólica - Exemplo

$$\frac{u_{i,j+1} - u_{i,j}}{h_y} - \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2} = 0$$

sujeita às seguintes condições

$$u(0, y_j) = u(1, y_j) = 0, \quad y > 0$$

e

$$u(x, 0) = \sin(\pi x), \quad 0 \leq x \leq 1$$

com $0 \leq j \leq 4$ e $1 \leq i \leq 1$.

EDP Parabólica - Exemplo

Basta resolver o sistema! Quantas variáveis tem?

$$\left\{ \begin{array}{l} \frac{u_{1,1} - u_{1,0}}{0.2} - \frac{u_{2,0} - 2u_{1,0} + u_{0,0}}{0.5^2} = 0 \\ \frac{u_{1,2} - u_{1,1}}{0.2} - \frac{u_{2,1} - 2u_{1,1} + u_{0,1}}{0.5^2} = 0 \\ \frac{u_{1,3} - u_{1,2}}{0.2} - \frac{u_{2,2} - 2u_{1,2} + u_{0,2}}{0.5^2} = 0 \\ \frac{u_{1,4} - u_{1,3}}{0.2} - \frac{u_{2,3} - 2u_{1,3} + u_{0,3}}{0.5^2} = 0 \\ \frac{u_{1,5} - u_{1,4}}{0.2} - \frac{u_{2,4} - 2u_{1,4} + u_{0,4}}{0.5^2} = 0 \end{array} \right.$$

EDP Parabólica - Exemplo

$$\left\{ \begin{array}{l} \frac{u_{1,1} - u_{1,0}}{0.2} - \frac{u_{2,0} - 2u_{1,0} + u_{0,0}}{0.5^2} = 0 \\ \frac{u_{1,2} - u_{1,1}}{0.2} - \frac{u_{2,1} - 2u_{1,1} + u_{0,1}}{0.5^2} = 0 \\ \frac{u_{1,3} - u_{1,2}}{0.2} - \frac{u_{2,2} - 2u_{1,2} + u_{0,2}}{0.5^2} = 0 \\ \frac{u_{1,4} - u_{1,3}}{0.2} - \frac{u_{2,3} - 2u_{1,3} + u_{0,3}}{0.5^2} = 0 \\ \frac{u_{1,5} - u_{1,4}}{0.2} - \frac{u_{2,4} - 2u_{1,4} + u_{0,4}}{0.5^2} = 0 \end{array} \right.$$

EDP Hiperbólica

Consideramos a equação da onda como exemplo de uma equação diferencial parcial hiperbólica.

Equação da Onda

$$u_{yy} - \alpha^2 u_{xx} = 0$$

com $0 \leq x \leq l$ e $y > 0$ que satisfaz as seguintes condições

$$\begin{aligned} u(x, 0) &= f(x), & u_y(x, 0) &= g(x) \\ u(0, y) &= 0, & u(l, y) &= 0, \text{ com } 0 \leq x \leq l \end{aligned}$$

Descobrir $u(x, y)$ usando o método das diferenças divididas!!!

EDP Hiperbólica

Discretizando e substituindo as derivadas de segunda ordem por uma aproximação de diferenças finitas

$$u_{xx}|_{x_i, y_j} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2}$$

$$u_{yy}|_{x_i, y_j} \approx \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_y^2}$$

com $u_{i,j} = u(x_i, y_j)$ para $1 \leq i \leq N_x - 1$ e $1 \leq j \leq N_y - 1$

EDP Hiperbólica

Substituímos na equação e obtemos

$$\frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_y^2} - \alpha^2 \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2} = 0$$

Sabemos $u_{i,0} = f(x_i)$, $u_{0,j} = 0$ e $u_{N_x,j} = 0$.
E também

$$\frac{u_{i,1} - u_{i,0}}{h_y} = g(x_i)$$

EDP Hiperbólica - Exemplo

Exemplo

Resolva o seguinte problema

$$u_{yy} - 4u_{xx} = 0$$

com $0 \leq x \leq 1$ e $y > 0$ que satisfaz as seguintes condições

$$\begin{aligned} u(x, 0) &= \sin(\pi x), & u_y(x, 0) &= 0 \\ u(0, y) &= 0, & u(1, y) &= 0, \text{ com } 0 \leq x \leq 1 \end{aligned}$$

A solução exacta

$$u(x, y) = \sin(\pi x) \cos(2\pi y).$$

Compare os resultados!

PDETOOL no MatLab

- Escrever 'pdetool' na linha de comandos do Matlab. Ajuste os domínios das variáveis (Options/Axes Limits).
- Defina uma região rectangular no domínio. Defina as condições de fronteira do problema (duplo clique em cada recta da fronteira).
- Caracterize a equação diferencial (pde/pde specification).
- Divida o domínio em triângulos tão pequenos quanto queira (botão ' Δ ').
- O botão '=' fornece a solução da equação através de um gráfico colorido bidimensional.
- O botão seguinte fornece um gráfico tridimensional com a solução da equação.

Referencias



J. Nocedal e S. Wright. Numerical Optimization. *Springer Series in Operations Research*. Springer. 1999.



S. G. Nash e A. Sofer. Linear and Nonlinear Programming. *Industrial Engineering Series*. McGraw-Hill International Editions. 1996.



M. S. Bazaraa, H. D. Sherali e C. M. Shetty. Nonlinear Programming Theory and Algorithms. John Wiley and Sons. 1993.